**May 12 Home questions**

#1. WAP to change the value of constant integer using pointers.

Code:

#include <stdio.h>

int main()

{

    const float pi\_285=3.14;

    float \*ptr\_285;

    int n;

    ptr\_285=&pi\_285;

    printf("The value of pi is %f\n",\*ptr\_285);

    printf("But its okay, give your own desired value to pi\n");

    scanf("%f",ptr\_285);

    printf("=-=-==-=-=-=-=-=-=-==-=-=-=-=-=-\n");

    printf("The new value you have given is = %f\n",\*ptr\_285);

    printf("=-=-==-=-=-=-=-=-=-==-=-=-=-=-=-\n");

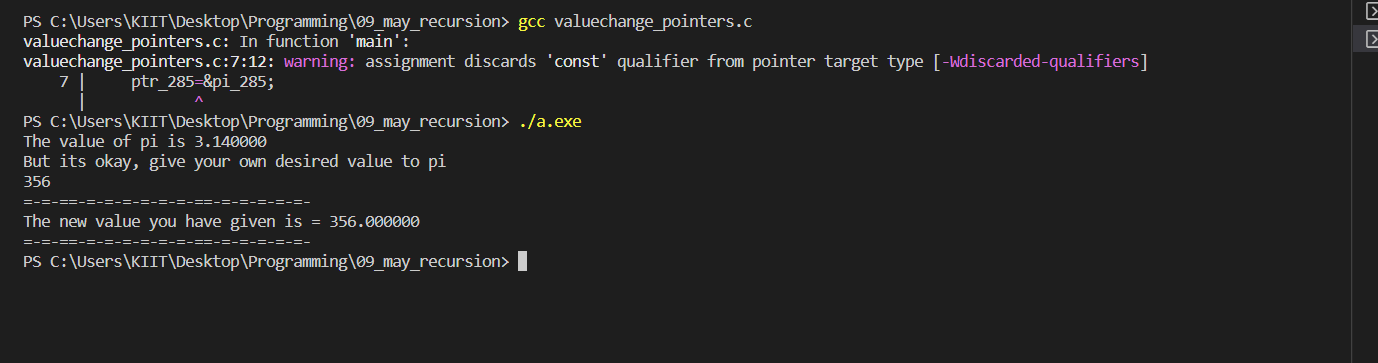
    n=1%10;

    printf("%d ",n);

    return 0;

}

Output:



#2. WAP to count distinct number of vowels and consonants present in a string

using pointer.

Code:

#include <stdio.h>

#include <string.h>

int main()

{

    char arr\_285[20], \*ptr\_285;

    int i,n,cCount\_285,vCount\_285;

    printf("Provide your string\n");

    fgets(arr\_285,20,stdin);

    ptr\_285=&arr\_285[0];

    n=strlen(arr\_285);

    while(\*ptr\_285!='\0')

    {

        if(\*ptr\_285=='U'|| \*ptr\_285=='O'|| \*ptr\_285=='I'|| \*ptr\_285=='E'|| \*ptr\_285=='A'|| \*ptr\_285=='u'|| \*ptr\_285=='o'|| \*ptr\_285=='i'|| \*ptr\_285=='e'|| \*ptr\_285=='a')

            vCount\_285++;

        ptr\_285++;

    }

    cCount\_285=n-vCount\_285-1;

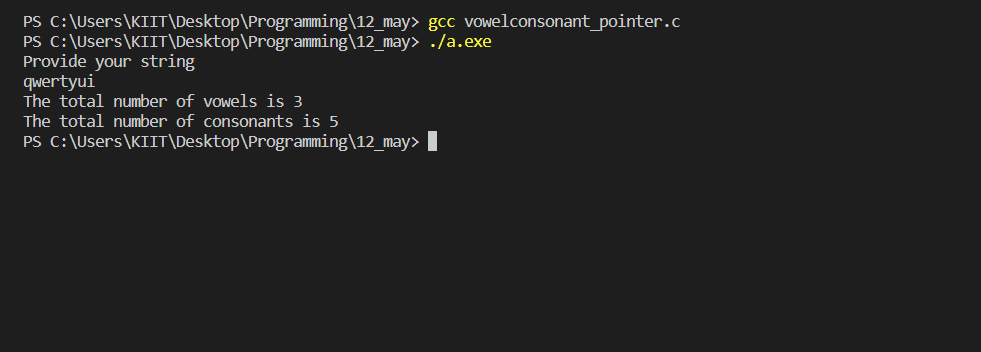
    printf("The total number of vowels is %d\n",vCount\_285);

    printf("The total number of consonants is %d\n",cCount\_285);

    return 0;

}

Output:



#3. WAP to print all permutations of a given string using pointers.

Code:

#include <stdio.h>

#include <string.h>

void changePosition(char \*ch1, char \*ch2)

    {

        char tmp;

        tmp = \*ch1;

        \*ch1 = \*ch2;

        \*ch2 = tmp;

    }

void charPerm(char \*cht, int startnum\_285, int endnum\_285)

    {

        int i;

        if (startnum\_285 == endnum\_285)

            printf("%s  ", cht);

        else

        {

            for (i = startnum\_285; i <= endnum\_285; i++)

            {

                changePosition((cht+startnum\_285), (cht+i));

                charPerm(cht, startnum\_285+1, endnum\_285);

                changePosition((cht+startnum\_285), (cht+i));

            }

        }

    }

int main()

    {

        char str\_285[4];

        printf("Input a 4 letter word: ");

        scanf("%s",str\_285);

        printf("\n Permutations of given string using pointers:\n");

        int n = strlen(str\_285);

        printf(" The permutations of the string are : \n");

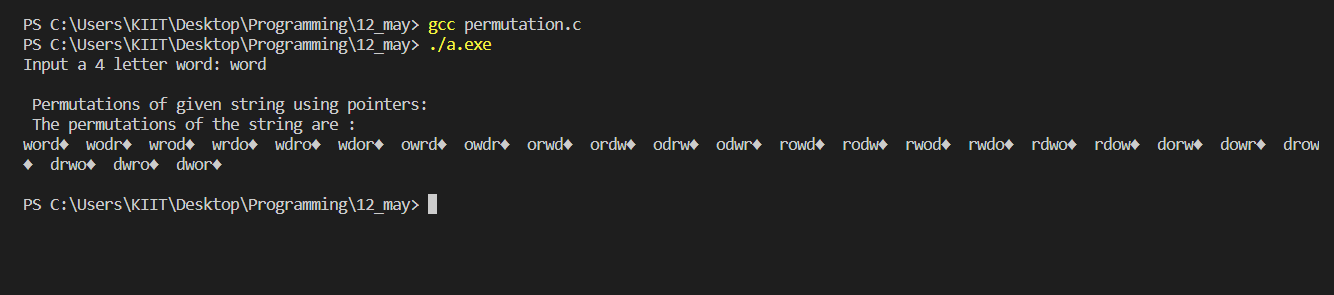
        charPerm(str\_285, 0, n-1);

        printf("\n\n");

        return 0;

    }

Output:



#4. WAP to swap two consecutive characters starting from left to right of a

string. (Hint: Take two pointers and increment by 2.)

Code:

#include <stdio.h>

#include <string.h>

void swap(char\* ptr1\_285, char\* ptr2\_285)

{

    char temp;

    temp = \*ptr1\_285;

    \*ptr1\_285= \*ptr2\_285;

    \*ptr2\_285= temp;

}

int main()

{

    int i,j,n;

    char \*ptr1\_285,\*ptr2\_285,string[20];

    printf("Please give a short string\n");

    gets(string);

    ptr1\_285=&string[0]; ptr2\_285=&string[1];

    n= strlen(string);

    for(i=0;i<n-1;i++)

    {

        swap(ptr1\_285,ptr2\_285);

        ptr1\_285++;

        ptr2\_285++;

    }

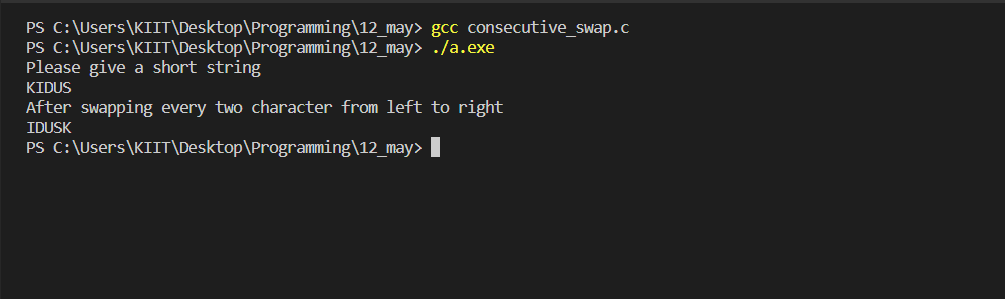
    printf("After swapping every two character from left to right\n");

    puts(string);

    return 0;

}

Output:



#5. WAP to find the number of times that a given word (i.e. a short string)

occurs in a sentence (i.e. a long string!).

Code:

#include <stdio.h>

#include <string.h>

int main()

{

    int i,j,count\_285,n;

    char arr\_285[30][30],sub\_285[30];

    printf("How many words does your sentence have?\n");

    scanf("%d",&n);

    printf("Please provide a sentence\n");

    for(i=0;i<n;i++)

    {

        scanf("%s",arr\_285[i]);

    }

    printf("What is the string you are looking to count?\n");

    scanf("%s",sub\_285);

    for(i=0;i<n;i++)

    {

        if(strcmp(arr\_285[i],sub\_285)==0)

        count\_285++;

    }

    printf("count=%d",count\_285);

    return 0;

}

Output:

